

KARTA PRZEDMIOTU

1. Informacje ogólne

Nazwa przedmiotu i kod (wg planu studiów):	Programowanie wieloplatformowe, D1_1
Nazwa przedmiotu (j. ang.):	Multiplatform programming
Kierunek studiów:	Informatyka
Specjalność/specjalizacja:	Informatyka praktyczna
Poziom kształcenia:	studia I stopnia
Profil kształcenia:	praktyczny (P)
Forma studiów:	studia stacjonarne
Obszar kształcenia:	nauki techniczne
Dziedzina:	nauki techniczne
Dyscyplina nauki:	informatyka
Koordinator przedmiotu:	dr Marcin Skuba

2. Ogólna charakterystyka przedmiotu

Przynależność do modułu:	specjalnościowego
Status przedmiotu:	obowiązkowy
Język wykładowy:	polski
Rok studiów, semestr:	II, 3
Forma i wymiar zajęć według planu studiów:	stacjonarne - wykład 30 h, ćw. laboratoryjne 30 h
Interesariusze i instytucje partnerskie (nieobowiązkowe)	
Wymagania wstępne / Przedmioty wprowadzające:	Podstawy programowanie w języku C/ Programowanie I,

3. Bilans punktów ECTS

Całkowita liczba punktów ECTS (wg planu studiów; 1 punkt =25-30 godzin pracy studenta, w tym praca na zajęciach i poza zajęciami): (A + B)	5	stacjonarne
A. Liczba godzin wymagających bezpośredniego udziału nauczyciela (kontaktowych, w czasie rzeczywistym, w tym testy, egzaminy etc) z podziałem na typy zajęć oraz całkowita liczba punktów ECTS osiągniętych na tych zajęciach	obecność na wykładach obecność na ćwiczeniach laboratoryjnych ćwiczenia projektowe udział w konsultacjach W sumie: ECTS	30 30 10 70 2,6
B. Poszczególne typy zadań do samokształcenia studenta (niewymagających bezpośredniego udziału nauczyciela) wraz z planowaną średnią liczbą godzin na każde i sumaryczną liczbą ECTS (np. praca w bibliotece, w sieci, na platformie e-learningowej, w laboratorium, praca nad projektem końcowym, przygotowanie ogólne; suma poszczególnych godzin powinna zgadzać się z liczbą ogólną)	przygotowanie ogólne opracowanie dokumentacji (sprawozdań) praca nad projektem studiowanie zalecanej literatury praca w sieci w sumie: ECTS	10 8 17 15 15 65 2,4
C. Liczba godzin praktycznych/laboratoryjnych w ramach przedmiotu oraz związana z tym liczba punktów ECTS (ta liczba nie musi być powiązana z liczbą godzin kontaktowych, niektóre zajęcia praktyczne/laboratoryjne mogą odbywać się bez udziału nauczyciela):	udział w zajęciach praca samodzielna w sumie: ECTS	30 30 60 2,2

4. Opis przedmiotu

Cel przedmiotu: Celem przedmiotu jest osiągnięcie odpowiedniej wiedzy i umiejętności dotyczących programowania w wysoko-poziomym, obiektowym języku Java, używanym pod kątem programowania wieloplatformowego.
Metody dydaktyczne: wykład - pokaz, laboratorium - zadania problemowe
Treści kształcenia (w rozbiciu na formę zajęć (jeśli są różne formy) i najlepiej w punktach): Wykłady: Wprowadzenie do programowania obiektowego. Kompilacja kodu źródłowego do B-kodu. Polecenia kompilacji, uruchamiania, budowy plików JAR - wykonywane na różnych platformach systemowych (Linux, Solaris, Windows). Mechanizmy Java wykorzystywane do budowy aplikacji (dziedziczenie, wielowątkowość). Strumienie danych przesyłanie między systemami operacyjnymi. Kodowanie i szyfrowanie danych. Tworzenie interfejsu użytkownika. Współpraca aplikacji z relacyjną bazą danych niezależnie od systemu operacyjnego. Zintegrowane środowisko programistyczne – Eclipse. Struktura aplikacji dla systemu android. Wykorzystanie poznanych mechanizmów Java do programowania prostych aktywności. Testowanie aplikacji na platformach systemowych innych od tych na których zostały napisane i skompilowane.

Ćwiczenia (audytoryjne/laboratoryjne/ projektowe, warsztaty itp):

Środowisko programistyczne – struktura programu, kompilacja i uruchamianie wieloplatformowe. Programowanie obiektowe – deklaracja klas, tworzenie obiektów, zmienne, metody. Dziedziczenie klas. Mechanizmy przysyłania, przeciążanie metod i konstruktorów. Budowa aplikacji – komponenty. Programowanie współbieżne. Strumienie danych. Przesyłanie informacji pomiędzy aplikacjami uruchomionymi na różnych platformach systemowych. Aplikacje bazodanowe – zdalne i lokalne. Biblioteki dedykowane – ogólnodostępne, własne i dodatkowe (komercyjne). Tworzenie prostych aktywności oraz zarządzanie danymi w OS Android.

5. Efekty kształcenia i sposoby weryfikacji

Efekty kształcenia				
Efekt przedmiotu	Student, który zaliczył przedmiot (spełnił minimum wymagań)			Efekt kierunkowy
D1_1_W01	Wiedza: 1. Student zna podstawowe elementy obiektowego języka Java. 2. Student zna mechanizmy gromadzenia danych oraz przesyłania ich pomiędzy aplikacjami wielu platform systemowych i sprzętowych. 3. Student wie jak programować aplikacje uruchamiane na wielu różnych systemach operacyjnych.			K_W06
D1_1_W02				K_W07
D1_1_W03				K_W18
D1_1_U01	Umiejętności 1. Student umie używać środowisk programistycznych do budowy aplikacji w języku Java. 2. Student potrafi napisać aplikację w środowisku obiektowym, która może być uruchamiana na kilku platformach systemowych. 3. Student potrafi stworzyć system wymiany danych pomiędzy aplikacjami pracującymi na wielu platformach.			K_U03
D1_1_U02				K_U10
D1_1_U03				K_U31
D1_1_K01	Kompetencje społeczne 1. Student rozumie potrzebę poznawania nowych narzędzi programistycznych wykorzystywanych w programowaniu wieloplatformowym. 2. Student rozumie potrzebę wykorzystania nabytej wiedzy na rynku programistycznym			K_K01
D1_1_K02				K_K08
Sposoby weryfikacji efektów kształcenia:				
Lp.	Efekt przedmiotu	Sposób weryfikacji	Ocena formująca	Ocena końcowa
1	D1_1_W01 D1_1_W02 D1_1_W02	Rozwiązywanie problemów dotyczących programowania wieloplatformowego	Oceny z odpowiedzi ustnej, oceny za aktywność, ocena z kolokwium	Średnia ocen formujących
2	D1_1_U01 D1_1_U02 D1_1_U03	Rozwiązywanie zadań problemowych na zajęciach laboratoryjnych, kolokwium	Oceny za projekt	Średnia ocen formujących
3	D1_1_K01 D1_1_K02	Obserwacja, pogadanka.	Oceny za aktywność	Średnia ocen formujących

Kryteria oceny		
w zakresie wiedzy		Efekt kształcenia
Na ocenę 3,0	Student uzyskał min. 50% wymaganej wiedzy w zakresie obowiązującego materiału. Student: - wie jak programować aplikacje wieloplatformowe na poziomie podstawowym, - wie jak programować obiektowo wykorzystując mechanizm dziedziczenia, metody biblioteczne, metody przechowywania danych oraz komponenty aplikacji na poziomie podstawowym, - zna w stopniu podstawowym środowisko do programowania aplikacji na OS Android oraz wie jak stworzyć prostą aktywność.	D1_1_W01 D1_1_W02 D1_1_W03
Na ocenę 5,0	Student zdobył powyżej 95% wymaganej wiedzy w zakresie obowiązującego materiału. Student uzyskał wiedzę na ocenę 3.0 oraz dodatkowo potrafi wykorzystać środowisko zintegrowane Eclipse do tworzenia wieloklasowych aplikacji uruchamianych na różnych platformach systemowych. Zna bardziej zaawansowane elementy projektu dla OS Android.	D1_1_W01 D1_1_W02 D1_1_W03
w zakresie umiejętności		
Na ocenę 3,0	Student uzyskał min. 50% wymaganych umiejętności w zakresie obowiązującego materiału. Student potrafi: - stworzyć aplikację w języku Java zawierającą własny interfejs użytkownika, przechowującą dane w pliku, - stworzyć projekt kilkuklasowy oraz przekazywać pomiędzy klasami dane, - zbudować prostą aplikację wg własnego lub narzuconego pomysłu, - wykorzystać poznany język Java do zbudowania aplikacji na innej platformie niż Java SE np. OS Android.	D1_1_U01 D1_1_U02 D1_1_U03
Na ocenę 5,0	Student uzyskał powyżej 95% umiejętności w zakresie obowiązującego materiału. Uzyskał umiejętności na ocenę 3.0 oraz dodatkowo: - umie zbudować aplikację przechowującą dane w relacyjnej bazie danych oraz umie wykorzystać zintegrowane środowisko programistyczne, - potrafi zmieniać strukturę projektu narzuconą przez środowisko Eclipse do realizacji własnego rozwiązania, - w szerokim stopniu programować w wielu platformach systemowych.	D1_1_U01 D1_1_U02 D1_1_U03
w zakresie kompetencji społecznych		
Na ocenę 3,0	Student osiągnął wymagane kompetencje społeczne na poziomie min. 50%.	D1_1_K01 D1_1_K02
Na ocenę 5,0	Student osiągnął wymagane kompetencje społeczne na poziomie wyższym niż 90%.	D1_1_K01 D1_1_K02

Zalecana literatura

Literatura podstawowa:	<ol style="list-style-type: none"> 1. Cay S. Horstmann, Java 8. Przewodnik doświadczonego programisty, Helion 2016 2. Eckel, Bruce, Thinking in Java, Helion, Gliwice 2006. 3. Darwin, Lan F, Android, Helion, Gliwice 2013, 4. Kubiak, Mirosław J. Java, Helion, Gliwice 2011, 5. Darwin, Lan F, Android, Helion, Gliwice 2013,
Literatura uzupełniająca:	<ol style="list-style-type: none"> 1. Mc Laughlin, Brett, Java i XML, Helion, Gliwice 2001. 2. Morgan, Michael, Poznaj język Java, Mikom, Warszawa 2001. 3. Rychlicki Wiesław, Programowanie w języku Java, Helion, Gliwice 2012

Informacje dodatkowe:

<p>Dodatkowe obowiązki prowadzącego wraz z szacowaną całkowitą liczbą godzin: <i>(np. indywidualne konsultacje, poprawa prac, przygotowanie projektu zaliczeniowego, egzaminu, przygotowanie ćwiczeń e-learningowych). Przykład poniżej</i></p>
Przygotowanie i aktualizacja wykładów, ćwiczeń i zadań domowych – 45 godzin
Ocena sprawozdań i zadań domowych – 10 godzin
Konsultacje – 20 godzin
W sumie: 75 godzin